UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/739,618	12/18/2000	John H. Howard	5181-59100	3682

58467 7590 05/14/2008

MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER	
DUONG, THOMAS	

ART UNIT	PAPER NUMBER
2145	

MAIL DATE	DELIVERY MODE
05/14/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/739,618
Filing Date: December 18, 2000
Appellant(s): HOWARD, JOHN H.

____
Howard
For Appellant

EXAMINER'S ANSWER

This is in response to the Appeal Brief filed January 17, 2008 appealing from the Office action mailed March 21, 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,724,581	Kozakura	03-1998
5,870,757	Fuller	02-1999
6,571,259	Zheng et al.	05-2003

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 2-3, 8, 12-15, 18, 22-23, 29-32, and 35 are rejected under 35 U.S.C. 102(b) as being anticipated by Kozakura (US005724581).

With regard to *claims 2, 8, 12, 22, and 29*, Kozakura discloses,

- *a non-volatile memory storing a first inode locating a first file in said storage and also storing a journal comprising a list of committed inodes; and* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 –

col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42)

Kozakura discloses, *"a current page table 2 is provided in the main storage unit and manages the position information in the data base storage unit concerning the latest physical page storing the latest updated data and the shadow physical page storing the data before the latest update"* (Kozakura, col.4, lines 41-45) and *"a current table management table 3 is provided in the main storage unit and manages as a shadow page table the current page table whose backup data are copied when a checkpoint is recorded, and manages the current page table updated after the checkpoint as the latest page table"* (Kozakura, col.4, lines 46-51). Hence, Kozakura teaches of the current and shadow page tables (i.e., Applicant's inodes) storing the position information (i.e., Applicant's locating) of the physical data (i.e., Applicant's first file) in the data base storage unit (i.e., Applicant's storage). In addition, Kozakura discloses, *"the present invention comprises a current page table for storing a page table in which a shadow page system manages a physical page corresponding to each logical page in a data base, and a current page table management table for managing the page table in the current page table using the shadow page system"* (Kozakura, col.3, lines 34-39). Hence, Kozakura teaches of the current page table management table (i.e., Applicant's journal) for managing the page tables (i.e., Applicant's inodes). Furthermore, Kozakura discloses, *"a non-volatile semiconductor memory such as a flash memory, a RAM disk, etc. can be used as the secondary storage unit 40"* (Kozakura, col.20, lines 28-30). Hence, Kozakura teaches of using non-volatile

memory to store the current and shadow page tables (i.e., Applicant's inodes), which, in turn, stores the position information of the physical data base data.

- *a block manager configured to copy said first inode to a second inode, wherein said block manager is configured to change said second inode in response to updates to said first file, and wherein said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory, whereby said second inode locates said first file in said storage, and wherein said block manager is configured to record said second inode in said journal.* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42) Kozakura discloses the first updating unit 7 *"[obtaining] a currently unused physical page, [copying] data in the latest page table to the physical page, and enters the copied data in the management table 3 as the latest page table for the logical page. The copied-from latest page table is entered in the current page table management table 3 as a shadow page table. Then, the newly-obtained physical page is set in the blank page management unit 6 as a physical page being used"* (Kozakura, col.5, lines 6-13), in response to *"when data on a logical page are updated as a result of an execution of a transaction"* (Kozakura, col.4, lines 66-67). In addition, Kozakura discloses, *"a second updating unit 8 [referring] to the current page table 2 which has been updated by the first updating unit 7, and writes the updated data on the logical page to the latest physical page corresponding to the logical page whose data are stored in the current page table 2 and should be updated. Then, it changes the position*

information pointing to the shadow page corresponding to the logical page in the current page table such that the information indicated the latest physical page" (Kozakura, col.5, lines 24-32). Hence, Kozakura teaches of a *"current page table for managing position information about [the] latest physical page storing latest updated data and a shadow physical page storing the data before [the] latest update"* (Kozakura, col.20, lines 54-57), a current page table management table for pointing to the latest page table as well as the shadow page table, and updating the table accordingly in response to execution of a transaction.

With regard to *claims 3, 18, 23, and 35*, Kozakura discloses,

- *wherein said commit of said first file comprises a commit command received from an external source which updates said first file.* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42)

With regard to *claims 13 and 30*, Kozakura discloses,

- *further comprising writing a master inode corresponding to an inode file including said second inode to a checkpoint record in said journal.* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42)

With regard to *claims 14-15 and 31-32*, Kozakura discloses,

- *wherein recovering from a system failure comprises:*

- *scanning said journal to locate a most recent checkpoint record and zero or more inodes subsequent to said most recent checkpoint record within said journal;* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42)
- *copying said master inode from said most recent checkpoint record to a volatile memory; and* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42)
- *updating an inode file corresponding to said master inode with said one or more inodes subsequent to said most recent checkpoint record.* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42)

- *wherein said updating said inode file comprises:*
    - *copying one or more blocks of said inode file storing said one or more inodes to a copied one or more blocks; and* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42)
    - *updating said master inode in said volatile memory to point to said copied one or more blocks.* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42)

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 4-5, 9-10, 19-20, and 24-25 are rejected under 35 U.S.C. 103(a) as being obvious over Kozakura (US005724581) and in view of Fuller (US005870757A).

With regard to *claims 4-5, 9-10, 19-20, and 24-25*, Kozakura discloses,

See claims 3, 8, 18, and 23 *rejection as detailed above.*

However, Kozakura does not explicitly disclose,

- *wherein said commit command comprises a file close command.*
- *wherein said commit command comprises an fsync command.*

Fuller teaches,

- *wherein said commit command comprises a file close command.* (Fuller, col.1, line 51 – col.3, line 48; col.22, line 35 – col.23, line 23)

  Fuller discloses of available transactional commands such as: 'close', 'fsync', 'read', 'write', 'commit', etc. that can cause the execution of a transaction.
- *wherein said commit command comprises an fsync command.* (Fuller, col.1, line 51 – col.3, line 48; col.22, line 35 – col.23, line 23)

  Fuller discloses of available transactional commands such as: 'close', 'fsync', 'read', 'write', 'commit', etc. that can cause the execution of a transaction.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Fuller with the teachings of Kozakura to provide a *"single transaction technique for journaling file systems … [that overcome] the performance degradation which may be experienced in conventional journaling file systems by entering each file system operation into the current active transaction"* (Fuller, col.1, lines 51-55). In addition, Fuller anticipates, *"in addition to increasing overall file system performance under even light computer system operational loads, even greater performance enhancement is experienced under relatively heavy load"* (Fuller, col.1, lines 59-62).

Claims 6-7, 16-17, 26-27, and 33-34 are rejected under 35 U.S.C. 103(a) as being obvious over Kozakura (US005724581) and in view of Zheng et al. (US006571259B1).

With regard to *claims 6-7 and 26-27*, Kozakura discloses,

See claims 2 and 22 *rejection as detailed above.*

However, Kozakura does not explicitly disclose,

- *wherein said journal further includes a checkpoint record including a description of an inode file, a block allocation bitmap, and an inode allocation bitmap.*
- *wherein the description comprises inodes for each of said inode file, said block allocation bitmap, and said inode allocation bitmap.*

Zheng teaches,

- *wherein said journal further includes a checkpoint record including a description of an inode file, a block allocation bitmap, and an inode allocation bitmap.* (Zheng, col.3, line 3 - col.4, line 14; col.13, line 66 – col.15, line 47)
- *wherein the description comprises inodes for each of said inode file, said block allocation bitmap, and said inode allocation bitmap.* (Zheng, col.3, line 3 - col.4, line 14; col.13, line 66 – col.15, line 47)

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Zheng with the teachings of Kozakura to provide an alternate method of *"[recovering] from a system failure by restoring the database to its consistent state existing just after commitment of the last completed transaction ... [by maintaining] a log file of the database changes and the commit commands ... [including] a sufficient amount of information (such as 'before' and 'after' images) in order to undo the changes made to the database since the last commit command"* (Zheng, col. 1, line 67 - col.2, line 8). In addition, according to Kozakura, *"the log of a transaction is stored in a log file in the secondary storage unit at the completion of the transaction. If a system failure occurs, the data base restores its state before the failure based on the page table management table and the page table stored as backup data in the secondary storage unit, and the log file"* (Kozakura, col.3, lines 44-49).

With regard to *claims 16-17 and 33-34*, Kozakura and Zheng disclose,

- *wherein said block map further comprises a first inode allocation bitmap indicating which inodes within said first inode file are allocated to files, the method further comprising:*

  - *copying said first inode allocation bitmap to a second inode allocation bitmap;* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42; Zheng, col.3, line 3 - col.4, line 14; col.13, line 66 – col.15, line 47)
  - *modifying said second inode allocation bitmap to reflect one or more inodes allocated to new files; and* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42; Zheng, col.3, line 3 - col.4, line 14; col.13, line 66 – col.15, line 47)
  - *establishing a third inode within said block map to said second inode allocation bitmap subsequent to said modifying said second inode bitmap.* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42; Zheng, col.3, line 3 - col.4, line 14; col.13, line 66 – col.15, line 47)
- *wherein said block map further comprises a first block allocation bitmap indicating which blocks within a storage including said block map are allocated to files, the method further comprising:*
  - *copying said first block allocation bitmap to a second block allocation bitmap;* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42; Zheng, col.3, line 3 - col.4, line 14; col.13, line 66 – col.15, line 47)

- *modifying said second block allocation bitmap to reflect one or more blocks allocated to files; and* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42; Zheng, col.3, line 3 - col.4, line 14; col.13, line 66 – col.15, line 47)
- *establishing a fourth inode within said block map to said second block allocation bitmap subsequent to said modifying said second block allocation bitmap.* (Kozakura, col.1, lines 40-50; col.2, lines 48-58, lines 18-26; col.3, lines 29-54; col.4, line 41 – col.5, line 41; col.6, line 58 – col.7, line 15; col.20, lines 23-34; col.20, line 48 – col.21, line 42; Zheng, col.3, line 3 - col.4, line 14; col.13, line 66 – col.15, line 47)

(10) Response to Argument

With regard to <u>*claims 2, 12, 22, and 29*</u>, the Applicant points out that:

- *The Final Office Action mailed March 21, 2007 ("Office Action") alleges that the inodes are taught in Kozakura as the page tables. Appellant respectfully disagrees. <u>Page tables locate physical pages stored in the memory system of the computer system, mapping logical pages used by the software to physical pages (see, e.g., Kozakura, col. 1, lines 32-28). Page tables that map logical pages to physical pages in memory have nothing to do with inodes that locate files on a storage.</u> Page tables cannot anticipate inodes, as they are completely different and are used for different purposes. Additionally, logical and physical pages are fixed in size (see, e.g., Kozakura col. 1, line 34), whereas files can have any size. Appellant notes that the standard for anticipation is one of fairly strict identity.*

Kozakura's page tables are significantly different from inodes, and cannot anticipate them.

However, the Examiner finds that the Applicants' arguments are not persuasive because Kozakura discloses, *"a current page table 2 is provided in the main storage unit and manages the position information in the data base storage unit concerning the latest physical page storing the latest updated data and the shadow physical page storing the data before the latest update"* (Kozakura, col.4, lines 41-45). Hence, Kozakura teaches of the current and shadow page tables (i.e., Applicant's inodes) storing the position information (i.e., Applicant's locating) of the physical data (i.e., Applicant's first file) in the data base storage unit (i.e., Applicant's storage).

With regard to *claims 2, 12, 22, and 29*, the Applicant points out that:

- *Kozakura teaches current page tables that locate the latest physical page storing the latest update and the shadow physical page storing data before the latest update. The current page table data structures cannot be a journal, since they are updated as transactions progress and thus do not comprise a list of committed inodes. Kozakura also teaches a backup page table that is created at a checkpoint. However, the backup page table is also not a list of committed inodes. Page tables merely map logical pages to physical pages. Thus, a backup page table created at a checkpoint is merely a snapshot of the current memory state. There is no list. Furthermore, the current state at checkpoint is merely the state of the current and shadow pages. These pages may generally comprise any combination of committed and uncommitted data, and thus are not committed inodes.*

However, the Examiner finds that the Applicants' arguments are not persuasive because the Examiner does not equate the *"current page table"* to the Applicant's journal. Instead, the Examiner equates Kozakura's *"current table management table 3"* to the Applicant's journal. Kozakura discloses, *"a current table management table 3 is provided in the main storage unit and manages as a shadow page table the current page table whose backup data are copied when a checkpoint is recorded, and manages the current page table updated after the checkpoint as the latest page table"* (Kozakura, col.4, lines 46-51) and *"the present invention comprises a current page table for storing a page table in which a shadow page system manages a physical page corresponding to each logical page in a data base, and a current page table management table for managing the page table in the current page table using the shadow page system"* (Kozakura, col.3, lines 34-39). Hence, Kozakura teaches of the current page table management table (i.e., Applicant's journal) for managing the page tables (e.g., shadow page table, current page table) (i.e., Applicant's inodes).

With regard to *claims 2, 12, 22, and 29*, the Applicant points out that:

- *Furthermore, Kozakura's checkpoints are not related to a commit command. Rather, they are performed cyclically or at a given time (Kozakura, col. 2, lines 48-49), or when no transactions are in progress (Kozakura, col. 2, lines 59-61). There is no command. Rather, Kozakura's system creates checkpoints automatically, without any command.*

However, the Examiner finds that the Applicants' arguments are not persuasive because Kozakura discloses the first updating unit 7 *"[obtaining] a currently unused*

physical page, [copying] data in the latest page table to the physical page, and enters the copied data in the management table 3 as the latest page table for the logical page. The copied-from latest page table is entered in the current page table management table 3 as a shadow page table. Then, the newly-obtained physical page is set in the blank page management unit 6 as a physical page being used" (Kozakura, col.5, lines 6-13), in response to *"when data on a logical page are updated as a result of an execution of a transaction"* (Kozakura, col.4, lines 66-67). In addition, Kozakura discloses, *"a second updating unit 8 [referring] to the current page table 2 which has been updated by the first updating unit 7, and writes the updated data on the logical page to the latest physical page corresponding to the logical page whose data are stored in the current page table 2 and should be updated. Then, it changes the position information pointing to the shadow page corresponding to the logical page in the current page table such that the information indicated the latest physical page"* (Kozakura, col.5, lines 24-32). Hence, Kozakura teaches of a *"current page table for managing position information about [the] latest physical page storing latest updated data and a shadow physical page storing the data before [the] latest update"* (Kozakura, col.20, lines 54-57), a current page table management table for pointing to the latest page table as well as the shadow page table, and updating the table accordingly in response to execution of a transaction.

With regard to *claims 2, 12, 22, and 29*, the Applicant points out that:

- *Claim 12 recites a combination of features including: "atomically updating said first file by establishing said second inode as the inode for said first file, wherein said establishing comprises storing said second inode in a journal stored in a*

nonvolatile memory". The same teachings of Kozakura highlighted above with respect to claim 2 are relied on to reject claim 12. Appellant respectfully submits that Kozakura does not anticipate the above highlighted features, either.

- *Claim 22 recites a combination of features including: "a non-volatile memory storing a first inode locating a first version of a file in said storage and also storing a journal comprising a list of committed inodes; and a block manager ... configured to atomically update the file, producing a second version of the file, in response to a commit of the file by writing said second inode to said non-volatile memory.., and wherein said block manager is configured to record said second inode in said journal". The same teachings of Kozakura highlighted above with respect to claim 2 are relied on to reject claim 22. Appellant respectfully submits that Kozakura does not anticipate the above highlighted features, either.*
- *Claim 29 recites a combination of features including: "atomically updating the file to the second version by establishing said second inode as the inode for the file, wherein said establishing comprises storing said second inode in a journal stored in a nonvolatile memory". The same teachings of Kozakura highlighted above with respect to claim 2 are relied on to reject claim 29. Appellant respectfully submits that Kozakura does not anticipate the above highlighted features, either.*

However, the Examiner finds that the Applicants' arguments are not persuasive because Kozakura discloses, *"a current page table 2 is provided in the main storage unit and manages the position information in the data base storage unit concerning the latest physical page storing the latest updated data and the shadow physical page storing the data before the latest update"* (Kozakura, col.4, lines 41-45) and *"a current table management table 3 is provided in the main storage unit and manages*

as a shadow page table the current page table whose backup data are copied when a checkpoint is recorded, and manages the current page table updated after the checkpoint as the latest page table" (Kozakura, col.4, lines 46-51). Hence, Kozakura teaches of the current and shadow page tables (i.e., Applicant's inodes) storing the position information (i.e., Applicant's locating) of the physical data (i.e., Applicant's first file) in the data base storage unit (i.e., Applicant's storage). In addition, Kozakura discloses, *"the present invention comprises a current page table for storing a page table in which a shadow page system manages a physical page corresponding to each logical page in a data base, and a current page table management table for managing the page table in the current page table using the shadow page system"* (Kozakura, col.3, lines 34-39). Hence, Kozakura teaches of the current page table management table (i.e., Applicant's journal) for managing the page tables (i.e., Applicant's inodes). Furthermore, Kozakura discloses, *"a non-volatile semiconductor memory such as a flash memory, a RAM disk, etc. can be used as the secondary storage unit 40"* (Kozakura, col.20, lines 28-30). Hence, Kozakura teaches of using non-volatile memory to store the current and shadow page tables (i.e., Applicant's inodes), which, in turn, stores the position information of the physical data base data.

With regard to *claims 8, 13, and 30*, the Applicant points out that:

- *Appellant respectfully submits that claim 8 recites a combination of features not taught or suggested in the cited art. For example, claim 8 recites a combination of features including: "said first inode is stored in an inode file, and wherein said inode file is identified by a master inode, and wherein said inode file is atomically*

updated with said second inode by writing said master inode subsequent to said commit command".

- *The Office Action relies on the same teachings highlighted above with regard to claim 2 to allegedly teach the features of claim 8. No teachings of Kozakura cited in the rejection have anything to do with the above features. Furthermore, as noted above, the Office Action alleges that Kozakura's page tables correspond to inodes. However, there is no page table file, identified by a master pare table, in Kozakura. Therefore, Kozakura cannot anticipate the above highlighted features of claim 8.*

However, the Examiner finds that the Applicants' arguments are not persuasive because Kozakura discloses, *"a current page table 2 is provided in the main storage unit and manages the position information in the data base storage unit concerning the latest physical page storing the latest updated data and the shadow physical page storing the data before the latest update"* (Kozakura, col.4, lines 41-45) and *"a current table management table 3 is provided in the main storage unit and manages as a shadow page table the current page table whose backup data are copied when a checkpoint is recorded, and manages the current page table updated after the checkpoint as the latest page table"* (Kozakura, col.4, lines 46-51). Hence, Kozakura teaches of the current and shadow page tables (i.e., Applicant's inodes) storing the position information (i.e., Applicant's locating) of the physical data (i.e., Applicant's first file) in the data base storage unit (i.e., Applicant's storage). In addition, Kozakura discloses, *"the present invention comprises a current page table for storing a page table in which a shadow page system manages a physical page corresponding to each logical page in a data base, and a current page table*

management table for managing the page table in the current page table using the shadow page system" (Kozakura, col.3, lines 34-39). Hence, Kozakura teaches of the current page table management table (i.e., Applicant's journal, master inode) for managing the page tables (i.e., Applicant's inodes). Furthermore, Kozakura discloses, *"a non-volatile semiconductor memory such as a flash memory, a RAM disk, etc. can be used as the secondary storage unit 40"* (Kozakura, col.20, lines 28-30). Hence, Kozakura teaches of using non-volatile memory to store the current and shadow page tables (i.e., Applicant's inodes), which, in turn, stores the position information of the physical data base data.

With regard to *claims 3, 18, 23, and 35*, the Applicant points out that:

- *Claims 3 and 23 depend from claims 2 and 22, respectively. Accordingly, the rejection of claims 3 and 23 is in error for at least the reasons highlighted above with regard to claims 2 and 22. Additionally, claims 3 and 23 recite combinations of features including: "said commit of said first file comprises a commit command received from an external source which updates said first file."*
- *Claims 18 and 35 depend from claims 12 and 29, respectively. Accordingly, the rejection of claims 18 and 35 is in error for at least the reasons highlighted above with regard to claims 12 and 29. Additionally, claims 18 and 35 recite combinations of features including: "said establishing said second inode is performed in response to a commit command."*

However, the Examiner finds that the Applicants' arguments are not persuasive because Kozakura discloses the first updating unit 7 *"[obtaining] a currently unused physical page, [copying] data in the latest page table to the physical page, and*

enters the copied data in the management table 3 as the latest page table for the logical page. The copied-from latest page table is entered in the current page table management table 3 as a shadow page table. Then, the newly-obtained physical page is set in the blank page management unit 6 as a physical page being used" (Kozakura, col.5, lines 6-13), in response to *"when data on a logical page are updated as a result of an execution of a transaction"* (Kozakura, col.4, lines 66-67). In addition, Kozakura discloses, *"a second updating unit 8 [referring] to the current page table 2 which has been updated by the first updating unit 7, and writes the updated data on the logical page to the latest physical page corresponding to the logical page whose data are stored in the current page table 2 and should be updated. Then, it changes the position information pointing to the shadow page corresponding to the logical page in the current page table such that the information indicated the latest physical page"* (Kozakura, col.5, lines 24-32). Hence, Kozakura teaches of a *"current page table for managing position information about [the] latest physical page storing latest updated data and a shadow physical page storing the data before [the] latest update"* (Kozakura, col.20, lines 54-57), a current page table management table for pointing to the latest page table as well as the shadow page table, and updating the table accordingly in response to execution of a transaction.

With regard to <u>*claims 4, 9, 19, and 24*</u>, the Applicant points out that:

- *Claims 4, 9, 19, and 24 depend from claims 3, 8, 18, and 23, respectively. Accordingly, the rejection of claims 4, 9, 19, and 24 is in error for at least the reasons highlighted above with regard to claims 3, 8, 18, and 23. Furthermore, the addition of Fuller to the rejection does not cure the deficiencies in the*

rejection of claims 3, 8, 18, and 23. Additionally, each of claims 4, 9, 19, and 24 recite a combination of features including: "said commit command comprises a file close command".

However, the Examiner finds that the Applicants' arguments are not persuasive because Fuller discloses of available transactional commands such as: 'close', 'fsync', 'read', 'write', 'commit', etc. that can cause the execution of a transaction. It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Fuller with the teachings of Kozakura to provide a *"single transaction technique for journaling file systems ... [that overcome] the performance degradation which may be experienced in conventional journaling file systems by entering each file system operation into the current active transaction"* (Fuller, col.1, lines 51-55). In addition, Fuller anticipates, *"in addition to increasing overall file system performance under even light computer system operational loads, even greater performance enhancement is experienced under relatively heavy load"* (Fuller, col.1, lines 59-62).

With regard to <u>*claims 5, 10, 20, and 25*</u>, the Applicant points out that:

- *Claims 5, 10, 20, and 25 depend from claims 3, 8, 18, and 23, respectively. Accordingly, the rejection of claims 5, 10, 20, and 25 is in error for at least the reasons highlighted above with regard to claims 3, 8, 18, and 23. Furthermore, the addition of Fuller to the rejection does not cure the deficiencies in the rejection of claims 3, 8, 18, and 23. Additionally, each of claims 5, 10, 20, and 25 recite a combination of features including: "said commit command comprises an fsync command".*

However, the Examiner finds that the Applicants' arguments are not persuasive because Fuller discloses of available transactional commands such as: 'close', 'fsync', 'read', 'write', 'commit', etc. that can cause the execution of a transaction. It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Fuller with the teachings of Kozakura to provide a *"single transaction technique for journaling file systems … [that overcome] the performance degradation which may be experienced in conventional journaling file systems by entering each file system operation into the current active transaction"* (Fuller, col.1, lines 51-55). In addition, Fuller anticipates, *"in addition to increasing overall file system performance under even light computer system operational loads, even greater performance enhancement is experienced under relatively heavy load"* (Fuller, col.1, lines 59-62).

With regard to *claims 6-7, 16-17, 26-27, and 33-34*, the Applicant points out that:

- *Claims 6 and 26 depend from claims 2 and 22, respectively. Accordingly, the rejection of claims 6 and 26 is in error for at least the reasons highlighted above with regard to claims 2 and 22. Furthermore, the addition of Zheng to the rejection does not cure the deficiencies in the rejection of claims 2 and 22. Additionally, each of claims 6 and 26 recite a combination of features including: "said journal further includes a checkpoint record including a description of an node file, a block allocation bitmap, and an inode allocation bitmap."*

However, the Examiner finds that the Applicants' arguments are not persuasive because Zheng discloses, *"Reserve indirect & data blocks without //by marking the CgBlkEntry's logging //active block bitmap Update in-memory file mapping data //no*

on-disk change at all structure Commit: GET_UPDATE_DESCRIPTOR(u,UPD_blockwrite); //get the UFS_Descriptor Traverse the in-memory file mapping data structure, commit necessary indirect & data blocks. For each indirect & data block, do the allocation logging: alloc lockWrite/newIndblkWrite/indblkWrite/inodeWrite In the traversal, test whether UFS_UpdateDescriptor transaction array overflow, do u->complete if necessary; Remove the committed block mappings from the in-memory mapping data structure; u->complete(FILE_OK); update inode Release/Abort: Traverse the in-memory file mapping data structure, release indirect/data blocks accordingly by marking the CgBlkEntry's active block bitmap. Remove the released block mappings from the data structure. Exported interfaces File Status getFileMap(ulong firstLogicalBlock, ulong size, extentEntry &extList, ByteIndex &fileSize); File Status allocSpace(ulong firstLogicalBlock, ulong size, extentEntry* &extList, ByteIndex &fileSize); File_Status commitAllocation(extentEntry* extList, ByteIndex lastOffset); File Status releaseAllocation(extentEntry* extList);"* (Zheng, col.14, line 46 – col.15, line 14).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Thomas Duong/

Examiner, Art Unit 2145

May 17, 2008

Conferees:

/Jason D Cardone/
Supervisory Patent Examiner, Art Unit 2145

/Rupal D. Dharia/
Supervisory Patent Examiner, Art Unit 2141